Listen and Chant Before You Read: The Ladder of Beauty in LM Pre-Training

Yoshinori Nomura
Mirage Mountain Technologies Inc.
nomura@miragemt.com

Abstract

We show that pre-training a Transformer on music before language significantly accelerates language acquisition. Using piano performances (MAESTRO dataset), a developmental pipeline—music → poetry → prose—yields a 17.5% perplexity improvement over random initialization ($p < 0.001$, 5 seeds), with music and poetry improving orthogonal model components (internal computation and embeddings, respectively). Convergence tests confirm that this is not a transient head start: at $d = 64$, multi-seed validation (5 seeds) shows a persistent 5.5% gap at plateau ($p = 0.017$), with the pipeline converging faster and to a lower loss in every run. Real music matches the transfer ceiling of synthetic patterns with one-third the data, and scaling experiments reveal that optimal pre-training data volume shifts with model capacity ($-3\% \rightarrow +3\% \rightarrow +6\%$ advantage of larger datasets from $d = 16$ to $d = 64$). These results establish a capacity-dependent data curation principle and suggest that structured human creative outputs provide an efficient pre-training substrate for language models.

1 Introduction

Standard language model pre-training learns directly from text corpora. Recent work has explored an additional stage *before* this: training first on non-linguistic data to establish general pattern recognition capabilities, then proceeding to language. We call this preliminary stage *pre-pre-training* (or *foundation warming*), following Lee et al. [2026], who demonstrated that training on synthetic patterns generated by Neural Cellular Automata (NCA)—two-dimensional discrete dynamical systems that produce spatially structured patterns—improved language model perplexity by up to 6% and accelerated convergence by 1.6×. This suggests that Transformers benefit from acquiring general sequence modeling capabilities before encountering language.

However, NCA patterns are two-dimensional cellular automata with limited structural similarity to language, which is inherently a one-dimensional sequential structure. We hypothesize that *music*—a sequential, hierarchically structured human creative output—provides a more natural pre-pre-training substrate for language models.

This hypothesis is grounded in cognitive science. Human infants acquire sensitivity to pre-linguistic regularities—rhythm, prosody, temporal patterns—before lexical content [Kuhl, 2004], and extensive evidence suggests that music and language share common cognitive substrates including hierarchical processing, long-range dependency tracking, and expectation-based computation [Patel, 2003, Koelsch, 2011]. We argue that acquiring a primitive sense of regularity prior to linguistic content is a fundamentally efficient

strategy, and that both biological and artificial learners converge on this solution: music, as a rich source of pre-linguistic order, should therefore be a superior pre-pre-training substrate.

We make six contributions:

1. We demonstrate that music pre-pre-training accelerates language learning by 26% in perplexity at the first epoch, persisting to 11.8% after convergence ($p < 0.001$, 5 seeds).
2. We show that data quality enables efficiency: real music by master composers reaches the same transfer ceiling as synthetic patterns with one-third the data volume.
3. We discover that adding poetry as an intermediate phase yields additive improvements (17.5% at epoch 2, $p < 0.001$), suggesting a developmental pipeline—music $\rightarrow$ poetry $\rightarrow$ prose—that parallels human language acquisition.
4. We identify the mechanism: music improves internal computation (attention + FFN), while poetry calibrates token embeddings toward language. These orthogonal contributions explain the additive effect.
5. We show that optimal pre-training data volume is a function of model capacity: the advantage of larger music datasets grows monotonically across three scales (d=16, 32, 64), revealing a capacity-dependent data curation principle.
6. We verify via convergence tests that the pipeline's advantage is not a transient head start: multi-seed validation at d=64 shows a persistent 5.5% gap at plateau ($p = 0.017$), with the pipeline converging faster and to a lower loss in every run.

2 Related Work

Pre-pre-training and foundation warming. The idea of training on auxiliary data before the main pre-training phase has emerged recently. Lee et al. [2026] introduced Neural Cellular Automata (NCA) as a source of synthetic pre-pre-training data, showing that non-linguistic pattern recognition transfers to language modeling. Their ablation study revealed that attention weights are the most transferable component, while MLP layers encode domain-specific statistics that can interfere with downstream learning. Our work extends this line of research by replacing synthetic 2D patterns with real 1D musical sequences, achieving substantially stronger transfer effects.

Music and Transformers. Music Transformer [Huang et al., 2019] demonstrated that self-attention can model long-range musical structure, and subsequent work has developed sophisticated MIDI tokenization schemes [Zeng et al., 2021]. However, these works focus on music generation; the transfer of musical representations to language tasks remains unexplored.

Cross-modal transfer learning. Transfer between modalities has been studied primarily in the vision-language direction [Lu et al., 2019]. Music-to-language transfer is, to our knowledge, novel. The closest precedent is the cognitive science literature on shared processing between music and language [Patel, 2003, Koelsch, 2011], which we translate into a concrete training methodology.

Music and language in cognitive science. Patel [2003] proposed the Shared Syntactic Integration Resource Hypothesis (SSIRH), arguing that music and language share neural resources for processing hierarchical structure. Koelsch [2011] documented overlapping neural substrates for processing expectations and violations in both domains. Kuhl [2004] showed that infants acquire sensitivity to pre-linguistic regularities—rhythm, prosody, temporal patterns—before lexical content, consistent with the view that acquiring a primitive sense of regularity prior to linguistic content is a fundamentally efficient strategy that both biological and artificial learners converge on.

3 Method

3.1 Model

We use a standard autoregressive Transformer decoder [GPT-2 architecture; Radford et al., 2019] with causal (left-to-right) self-attention and learned absolute position embeddings. We deliberately choose small dimensions to study learning efficiency under capacity constraints: $d_{\text{model}} = 16$ (the internal representation width), $n_{\text{heads}} = 1$ (single attention head), $d_{\text{head}} = 16$ (per-head dimension, equal to d_{model}), $n_{\text{layers}} = 8$ (Transformer blocks), $d_{\text{ff}} = 64$ (feedforward hidden dimension, $4 \times d_{\text{model}}$), yielding approximately 33K trainable parameters. We verify our findings at larger scales ($d = 32$, $d = 64$) in Section 4.4.

3.2 Music Tokenization

MIDI (Musical Instrument Digital Interface) files encode musical performances as sequences of discrete note events, each specified by pitch, onset time, duration, and velocity (loudness). We convert these events into a flat token sequence using a simplified version of the REMI (REvamped MIDI-derived) tokenization [Huang and Yang, 2020], which represents music as a linear sequence of tokens organized bar by bar.

Our vocabulary consists of 160 tokens in five categories:

- **Special tokens** (4): PAD (padding), BOS (beginning of sequence), EOS (end of sequence), BAR (bar boundary marker)
- **Position** (16): position within a bar on a 16th-note grid (0–15), indicating *when* a note occurs within the bar
- **Pitch** (128): MIDI pitch values (0–127, where 60 = middle C), indicating *which* note is played
- **Duration** (8): note length in 16th-note units (1–8), indicating *how long* the note sustains
- **Velocity** (4): dynamics bins (pp, p, f, ff), indicating *how loudly* the note is played

Each note event is represented by exactly four tokens (Position, Pitch, Duration, Velocity) in a fixed order, producing a deterministic *token grammar*—a set of strict ordering constraints on which token types can follow which: BAR $\rightarrow$ POS $\rightarrow$ PITCH $\rightarrow$ DUR $\rightarrow$ VEL $\rightarrow$ (POS | BAR | EOS). This grammar means the model must learn both the local syntax (the fixed ordering of token types within a note) and the musical content (which pitches, durations, and dynamics to predict).

3.3 Datasets

All sequential data is divided into *chunks*: fixed-length subsequences of seq_len $+ 1 = 257$ tokens, where the first 256 tokens serve as input and the last 256 as the prediction target (shifted by one position). Chunks are created by concatenating all source material and splitting into non-overlapping windows. The number of chunks thus determines the effective dataset size.

Table 1: Datasets used in our experiments.

Dataset	Source	Chunks	Nature
Synthetic music	Algorithmic generation	varied	Rule-based patterns
MAESTRO v2	Piano performances	36,061	58 composers, master pianists
Gutenberg Poetry	Project Gutenberg	36,000	Classical English poetry
WikiText-103	Wikipedia	—	General English prose

MAESTRO. The MAESTRO dataset [Hawthorne et al., 2019] contains aligned MIDI and audio from piano performances by professional pianists, covering composers from Bach to Rachmaninoff (58 composers, 1,276 pieces). We use the note-level annotations (pitch, onset time, offset time, velocity) and tokenize directly without relying on MIDI files.

Synthetic music. As a baseline data source, we generate synthetic music via a rule-based algorithm using the same MIDI token vocabulary as MAESTRO. For each piece, the generator (i) selects a random root note (C3–C5) and scale (major, minor, or pentatonic), (ii) creates a short motif of 1–2 bars by placing 2–6 notes at random positions within a 16th-note grid, and (iii) extends the piece to 4–16 bars by sampling from four operations: exact repetition (40%), transposition by a diatonic interval (20%), pitch variation where 30% of notes are replaced with scale-compatible alternatives (25%), or generation of a new contrasting phrase (15%). The resulting sequences exhibit surface-level musical structure—repetition, scale-constrained pitch, and simple variation—but lack the harmonic progressions, voice leading, phrase-level tension–resolution arcs, and long-range structural planning found in composed music. This design is intentionally minimal: it serves as a controlled baseline to isolate the contribution of *musical quality* (i.e., the structural richness present in real performances) from mere exposure to structured non-linguistic sequences.

Gutenberg Poetry Corpus. The corpus [BIG LAM, 2022] contains 3 million lines of English poetry from Project Gutenberg, tokenized with the GPT-2 tokenizer (vocabulary size 50,257). We subsample to 36,000 chunks to match the MAESTRO data size.

WikiText-103. We use a 10% subsample of WikiText-103 [Merity et al., 2017] for language evaluation, tokenized with the GPT-2 tokenizer.

3.4 Training Pipeline

We define a *developmental pipeline* (Figure 1): a sequence of training phases on progressively more language-like data, where each phase builds on the representations learned in the previous one. The term "developmental" is inspired by the trajectory of human language acquisition, which progresses from sub-linguistic

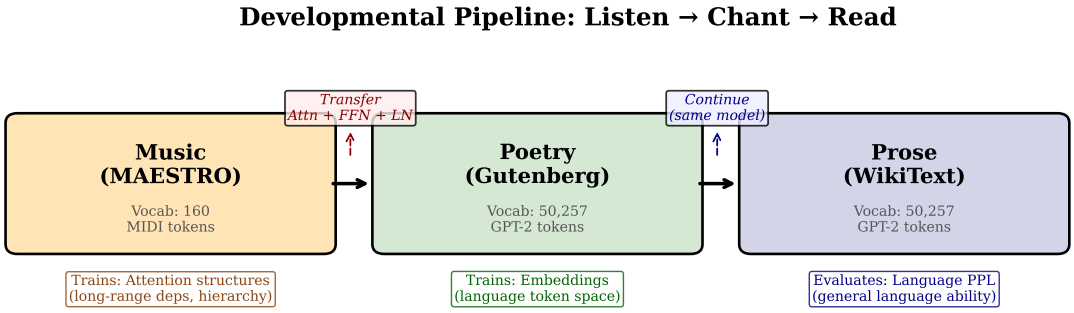

Figure 1: The developmental pipeline. Music pre-training establishes attention structures (long-range dependency tracking, hierarchical pattern recognition); poetry calibrates token embeddings toward the language space; prose training evaluates general language modeling ability. The vocabulary change between music (160 tokens) and poetry/prose (50,257 tokens) requires selective weight transfer (Section 3.4).

pattern recognition (rhythm, prosody) through structured language (nursery rhymes, songs) to general prose comprehension.

Our pipeline consists of three phases:

1. **Music phase**: Train a Transformer on music tokens (vocabulary size 160) until convergence. Training runs for up to 200 epochs with *early stopping*: if validation loss does not improve for 20 consecutive epochs (the *patience* parameter), training halts and the best checkpoint is retained.

2. **Poetry phase**: Construct a new model with a language vocabulary (GPT-2 tokenizer, 50,257 tokens) and transfer the learned weights via *selective weight transfer* (described below). Train on poetry for 3 epochs.

3. **Prose phase**: Continue training the same model on WikiText-103 for 3 epochs. Evaluate using *perplexity* (PPL), defined as $\text{PPL} = \exp(\mathcal{L})$ where $\mathcal{L}$ is the mean cross-entropy loss per token on the validation set. Lower perplexity indicates better language modeling: intuitively, it measures how many tokens the model is "choosing among" at each prediction step.

Selective weight transfer. When transitioning between phases with different vocabularies (music → poetry, or music → prose), we face a vocabulary mismatch: the music model uses 160 tokens while the language model uses 50,257. Our solution is to partition the model's parameters into two groups:

- **Internal computation layers**—attention weights (W_Q, W_K, W_V, W_O), feedforward weights (W_1, W_2), and layer normalization parameters (γ, β)—are *transferred*. These layers implement domain-general computation: dependency tracking, pattern recognition, and representation transformation. Their dimensions depend only on d_{model} and are identical across vocabularies.

- **Vocabulary-dependent layers**—the token embedding matrix and the output projection (language model head)—are *reinitialized* randomly. These layers map between the token space and the internal representation space, and their dimensions depend on vocabulary size. Since music tokens (note events) and language tokens (subwords) have no correspondence, these weights cannot be meaningfully transferred.

This design is motivated by Lee et al. [2026]'s finding that attention weights are the primary carriers of transferable computation, while MLP weights encode domain-specific statistics. A practical consequence is that immediately after transfer, the model's perplexity is near-random (∼50,000) because the embeddings carry no information; the transferred internal weights accelerate *how quickly* the model learns, not where it starts.

3.5 Experimental Design

Our experiments are organized in three phases, each addressing a distinct question.

3.5.1 Phase 1: Data Volume Control (Apples-to-Apples Comparison)

A naïve comparison of synthetic music (2,881 chunks) against MAESTRO (36,061 chunks) confounds data *quality* with data *quantity*. To disentangle the two, we control data volume at three levels (3k, 12k, and 36k chunks) and compare synthetic vs. MAESTRO at each level. This yields six pre-training conditions plus a random initialization baseline:

Table 2: Phase 1: Data volume-controlled conditions. Each condition trains a $d = 16$ music model until convergence (patience 20), then transfers internal weights to a language model and evaluates on WikiText-103 (10% subsample) for 3 epochs.

Condition	Source	Chunks	Purpose
Random	—	—	Baseline (no music pre-training)
Synth-3k	Synthetic	3,000	
Synth-12k	Synthetic	12,000	Quality effect at matched volume
Synth-36k	Synthetic	36,000	
MAESTRO-3k	MAESTRO subsample	3,000	
MAESTRO-12k	MAESTRO subsample	12,000	Quality effect at matched volume
MAESTRO-36k	MAESTRO (full)	36,000	

If MAESTRO outperforms synthetic data at matched volume, we can attribute the difference to data quality—the structural richness of composed music—rather than quantity. A secondary question is how pre-training data volume interacts with model capacity: does the optimal amount saturate at a level determined by the model's ability to absorb structure?

3.5.2 Phase 2: Statistical Validation (Multi-Seed)

Phase 1 uses a single seed (seed = 42). To establish statistical significance, we re-run four key conditions across five seeds (42, 123, 456, 789, 1024):

Table 3: Phase 2: Multi-seed conditions. Music checkpoints are shared (seed 42); only the language learning phase varies by seed.

Condition	Pipeline	Purpose
A	Random → Prose	Baseline
B	MAESTRO-12k → Prose	Best single-source from Phase 1
C	MAESTRO-36k → Poetry → Prose	Full developmental pipeline
D	Synth-36k → Prose	Data quality control

Music checkpoints are fixed across seeds: the stochasticity that matters is in language learning (random initialization of embeddings and LM head, data shuffling). We report mean ± std and paired t-tests for each condition vs. random baseline.

Note that condition B uses MAESTRO-12k rather than MAESTRO-36k. This choice is informed by Phase 1's finding that MAESTRO-12k achieves the best transfer at $d=16$ (Section 4.2), suggesting that the small model saturates before exhausting the full 36k dataset. Condition C retains MAESTRO-36k for the music phase because the poetry phase provides an additional pathway for the model to exploit the richer pre-training.

Compute-matched control. The developmental pipeline (condition C) involves more total training steps than the baseline: 3 epochs of poetry (∼6,075 batches) followed by 3 epochs of prose (∼8,526 batches), totaling ∼14,600 batches, versus ∼8,500 batches for the 3-epoch prose-only conditions. To rule out the possibility that the pipeline's advantage is simply due to additional compute, we include a *compute-matched* control: random initialization trained on WikiText-103 for 5 epochs (∼14,210 batches), matching the pipeline's total training budget to within 3%.

3.5.3 Phase 3: Scale × Data-Size Interaction

Phase 1 revealed a surprising result: at $d=16$, MAESTRO-12k *outperforms* MAESTRO-36k. We hypothesize that this reflects model capacity saturation—the 33K-parameter model cannot absorb more structure than 12k chunks provide. If so, larger models should shift the optimal data size upward.

To test this, we run three conditions (random, MAESTRO-12k, MAESTRO-36k) at three model scales:

Table 4: Phase 3: Scale experiment design. All models use 8 layers; $d=16$ reuses Phase 1 checkpoints.

d_{model}	Heads	d_{ff}	Params	Key question
16	1	64	33K	Phase 1 reuse (12k > 36k)
32	2	128	130K	Does 36k begin to overtake 12k?
64	4	256	400K	Does the reversal strengthen?

This design tests whether optimal pre-training data volume scales with model capacity—a question with practical implications for data curation at larger scales. Lee et al. [2026] showed that the optimal *complexity* of NCA patterns varies by target domain (low complexity for code, high for math). Our experiment extends this finding to a second axis: not just *what kind* of data to use, but *how much*, as a function of the model's capacity to absorb it.

3.5.4 Shared Hyperparameters

All conditions across all phases use identical hyperparameters: learning rate 10^{-3} with cosine decay to 10^{-4}, 200-step linear warmup, AdamW optimizer with weight decay 0.1, gradient clipping at 1.0, and gradient accumulation over 2 steps. Music models train for up to 200 epochs with early stopping (patience 20). Language evaluation uses 3 epochs on a 10% subsample of WikiText-103. The poetry phase, when present, consists of 3 epochs on 36,000 chunks of Gutenberg poetry. Phases 1 and 3 use a fixed random seed (seed = 42); Phase 2 varies the seed across five values (42, 123, 456, 789, 1024) for statistical validation.

4 Results

4.1 What the Music Model Learns

Before examining transfer effects, we characterize what the $d = 16$ model learns from music data.

Token grammar. The model learns the deterministic token grammar nearly perfectly: after BAR, it predicts POS with 97.8% probability; after POS, PITCH with 99.7%; after PITCH, DUR with 99.9%; after DUR, VEL with 99.8%.

Pattern completion. When presented with a motif (C–E–G) repeated three times, the model predicts BAR (pattern continuation) with 70.2% probability, and after BAR, predicts the correct starting position with 89.4% probability. This demonstrates genuine pattern recognition beyond token-level statistics.

Attention specialization. The single attention head devotes 89.5% of its attention mass to positions more than 8 tokens away, indicating specialization for long-range dependency tracking. Local patterns (the token grammar) are handled by the feedforward layers and embeddings.

4.2 Phase 1: Data Quality at Controlled Volume

Table 5 shows the Phase 1 results: WikiText-103 perplexity after 3 epochs of language learning, for each pre-training condition at matched data volumes.

Table 5: Phase 1 results: WikiText-103 validation perplexity at epoch 2 (final). Percentage shows improvement vs. random baseline.

Condition	E0	E1	E2
Random (baseline)	695.1	484.7	421.5
Synth-3k	611.6	456.4	401.5 (−4.7%)
Synth-12k	525.1	418.8	383.9 (−8.9%)
Synth-36k	494.3	398.3	370.7 (−12.1%)
MAESTRO-3k	603.5	452.4	403.9 (−4.2%)
MAESTRO-12k	514.7	404.7	**366.7 (−13.0%)**
MAESTRO-36k	503.8	404.7	375.4 (−10.9%)

Three findings emerge. First, all pre-training conditions outperform the random baseline, confirming that music pre-pre-training is beneficial regardless of data source or volume.

Second, **MAESTRO-12k achieves the best overall transfer** (PPL 366.7), surpassing even Synth-36k (PPL 370.7) with one-third the data volume. This provides direct evidence that data *quality*—the structural richness of real performances—can compensate for data quantity.

Third, **MAESTRO shows non-monotonic scaling**: performance improves from 3k to 12k but *degrades* from 12k to 36k (366.7 → 375.4). Synthetic data, by contrast, improves monotonically (401.5 → 383.9 → 370.7). We interpret this as capacity saturation: the 33K-parameter model can absorb the structural patterns in approximately 12k chunks of real music; additional data introduces redundancy or noise that slightly degrades transfer. Synthetic data, being structurally simpler, continues to benefit from additional examples because the model has not yet extracted all learnable patterns. Phase 3 (Section 4.4) tests this hypothesis by examining whether larger models shift the optimal data volume upward.

4.3 Phase 2: Statistical Validation

Table 6 reports the multi-seed results. All four pre-training conditions are evaluated across five random seeds; the music checkpoints are shared (seed 42) and only the language learning phase varies.

Table 6: Phase 2 results: WikiText-103 validation perplexity (mean ± std over 5 seeds). Δ is improvement vs. random baseline at epoch 2. All conditions vs. random are significant at $p < 0.001$ (paired t-test, $n = 5$).

Condition	E0	E1	E2	Δ	p
Random (baseline)	694.1 ± 17.6	483.4 ± 7.7	423.0 ± 5.3	—	—
MAESTRO-12k → Prose	512.5 ± 2.5	407.3 ± 2.4	373.2 ± 3.8	-11.8%	< 0.001
Synth-36k → Prose	499.4 ± 6.2	402.5 ± 4.7	371.5 ± 3.5	-12.2%	< 0.001
MAESTRO → Poetry → Prose	415.7 ± 4.4	369.5 ± 4.2	$\mathbf{349.0 \pm 5.8}$	$\mathbf{-17.5\%}$	< 0.001

Three results stand out.

First, **the findings from Phase 1 replicate with high consistency**. Standard deviations are small (3–6 PPL points), and all three pre-training conditions significantly outperform the random baseline ($p < 0.001$ for all comparisons, paired t-test, $n = 5$).

Second, **MAESTRO-12k and Synth-36k are statistically indistinguishable** at epoch 2 (373.2 vs. 371.5; $t = 0.97$, $p = 0.39$). This is notable: MAESTRO achieves comparable performance with one-third the data volume, but the advantage is not statistically significant at this sample size. The qualitative advantage of real music identified in Phase 1 (Section 4.2) is thus better characterized as an *efficiency* advantage—fewer data points needed to reach the same transfer effect—rather than a ceiling advantage.

Third, **the developmental pipeline achieves the strongest transfer by a wide margin**. The MAESTRO → Poetry → Prose condition (PPL 349.0) outperforms the next-best condition (Synth-36k, PPL 371.5) by 6.1%, and this difference is highly significant (vs. MAESTRO-12k: $t = 10.86$, $p < 0.001$). The poetry phase provides an additional 5.7 percentage points of improvement beyond music alone (17.5% vs. 11.8% relative to baseline), and this increment is consistent across all five seeds.

The epoch-0 perplexity reveals the mechanism behind the poetry phase's advantage. The random baseline starts at PPL 694±18; direct-transfer conditions (MAESTRO-12k, Synth-36k) start at ∼500—lower but still high, because the reinitialized embeddings carry no language-specific information. The poetry pipeline starts at 416±4, already below the random baseline's *final* performance after three full epochs of language learning. This dramatic head start reflects the fact that the poetry phase has already adjusted the embeddings toward the language token space, so the model enters the prose phase with both trained internal weights *and* partially calibrated embeddings.

Compute-matched control. The developmental pipeline uses more total training steps than the 3-epoch prose-only conditions (∼14,600 vs. ∼8,500 batches). To rule out that the pipeline's advantage is simply due to additional compute, we train a randomly initialized model on WikiText-103 for 5 epochs (∼14,210 batches), matching the pipeline's total budget to within 3%. Table 7 shows the results.

The compute-matched baseline (PPL 367.3±2.3) substantially outperforms the 3-epoch random baseline (423.0), confirming that additional training helps. However, the developmental pipeline (PPL 349.0 ± 5.8) still significantly outperforms the compute-matched control ($t = 5.47$, $p = 0.005$, paired t-test, $n = 5$). The pipeline's 17.5% improvement over the 3-epoch baseline cannot be attributed to additional compute alone: at matched compute budget, the pipeline retains a 5.0% advantage over pure prose training, demonstrating that the *content* of the developmental stages—not merely their duration—drives the transfer effect.

Table 7: Compute-matched control: WikiText-103 validation perplexity (mean ± std over 5 seeds). The compute-matched baseline trains for 5 prose epochs (∼14,210 batches) vs. the pipeline's ∼14,600 batches (music + poetry + prose).

Condition	Batches	Final PPL	p (vs. pipeline)
Random (3 ep prose)	∼8,500	423.0 ± 5.3	—
Compute-matched (5 ep prose)	∼14,210	367.3 ± 2.3	—
MAESTRO → Poetry → Prose	∼14,600	$\mathbf{349.0 \pm 5.8}$	0.005

4.4 Phase 3: Scale × Data-Size Interaction

Table 8 shows the Phase 3 results: WikiText-103 perplexity at epoch 2 for each condition at three model scales.

Table 8: Phase 3 results: WikiText-103 validation perplexity at epoch 2. Δ_R is improvement vs. random baseline at the same scale. $\Delta_{12/36}$ compares 36k against 12k (positive = 36k is better).

Scale	Random	MAESTRO-12k (Δ_R)	MAESTRO-36k (Δ_R)	$\Delta_{12/36}$
d=16 (33K)	418.9	364.3 (−13.0%)	375.4 (−10.4%)	−3.1%
d=32 (130K)	263.3	222.2 (−15.6%)	215.0 (−18.4%)	+3.3%
d=64 (400K)	167.2	149.1 (−10.8%)	**140.0 (−16.3%)**	**+6.1%**

The d=16 row differs slightly from Table 5 because Phase 3 re-runs the language transfer on a different machine; the qualitative pattern (12k > 36k) is identical.[1]

The results confirm the capacity saturation hypothesis and reveal a striking monotonic trend. At d=16, MAESTRO-12k outperforms 36k by 3.1%—the model is too small to absorb the additional structure in the larger dataset. At d=32, the relationship reverses: 36k overtakes 12k by 3.3%. At d=64, the advantage of 36k widens further to 6.1%.

Figure 2 visualizes these results, and Figure 3 shows the corresponding learning curves. The trajectory of the $\Delta_{12/36}$ column (−3.1% → +3.3% → +6.1%) is monotonically increasing, indicating that the advantage of larger pre-training datasets grows systematically with model capacity. Meanwhile, the 12k condition shows diminishing returns at larger scales (−13.0% → −15.6% → −10.8%), consistent with insufficient data for the model's capacity. The 36k condition, by contrast, maintains strong and stable transfer (−10.4% → −18.4% → −16.3%), and the continued expansion of the 36k–12k gap at $d = 64$ suggests that even 36k chunks may be insufficient at this scale—the optimal data volume likely exceeds 36k for 400K-parameter models.

These results extend Lee et al.'s finding that the optimal *complexity* of pre-training data varies by target domain to a second axis: the optimal *volume* of pre-training data varies by model capacity.

4.5 Convergence Test: Does the Gap Survive Long-Term Training?

Phases 1–3 evaluate language learning at a fixed three-epoch horizon. A natural concern is that the pipeline's advantage reflects a *head start* that vanishes once the random baseline trains long enough. To address this, we train both conditions—random initialization and the full developmental pipeline—until plateau (defined as < 2.5% improvement per epoch, with a minimum of 2 epochs before triggering). Each scale uses the best

[1] Phase 1 random baseline: 421.5; Phase 3: 418.9. The difference (< 1%) reflects numerical non-determinism across hardware and CUDA versions.

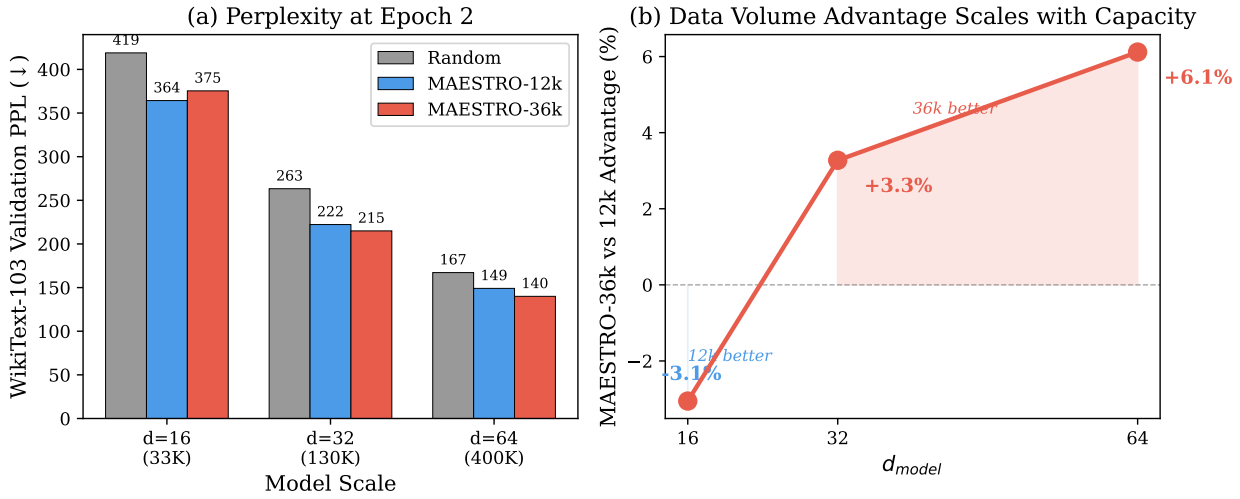

Figure 2: Phase 3: Scale × data-size interaction. **(a)** WikiText-103 validation perplexity at epoch 2 across three model scales. Music pre-training consistently improves over the random baseline at all scales, and the improvement grows with model size. **(b)** The advantage of MAESTRO-36k over MAESTRO-12k increases monotonically with model capacity ($-3.1\% \rightarrow +3.3\% \rightarrow +6.1\%$), confirming the capacity saturation hypothesis: small models cannot absorb large datasets, but larger models increasingly benefit from more data.

music data size identified in Phase 3: MAESTRO-12k for $d=16$ and MAESTRO-36k for $d=64$. To establish statistical reliability at the larger scale, we run the $d=64$ convergence test across five seeds (42, 123, 456, 789, 1024), matching the Phase 2 validation protocol.

Single-seed pilot ($d=16$). At $d=16$ (seed 42), the random baseline converges to PPL 346.8 after 7 epochs, while the pipeline reaches 322.5 in just 5 epochs—a 7.0% gap. This initial result motivated the multi-seed validation at $d=64$.

Multi-seed validation ($d=64$). Table 9 reports the per-seed results.

Table 9: Convergence test at $d=64$: training to plateau (5 seeds). The pipeline uses MAESTRO-36k + 3 epochs of poetry before prose. Gap is the percentage improvement of pipeline over random at convergence.

Seed	Random PPL (ep)	Pipeline PPL (ep)	Gap
42	122.0 (8)	112.9 (7)	−7.5%
123	117.9 (9)	116.1 (6)	−1.5%
456	118.3 (9)	114.9 (6)	−2.9%
789	122.3 (8)	111.4 (7)	−8.9%
1024	117.8 (9)	109.8 (8)	−6.8%
Mean ± std	119.7 ± 2.3	**113.0 ± 2.6**	**−5.5%**

The pipeline outperforms the random baseline in all five seeds, with a mean gap of 5.5% ($t = 3.90$, $p = 0.017$, paired t-test, $n = 5$). Three results emerge.

First, **the gap does not close**. Across all five seeds, the random baseline cannot reach the pipeline's converged perplexity even with additional epochs (8–9 epochs for random vs. 6–8 for pipeline). This rules out the "catch-up" hypothesis: the pipeline's advantage is not merely an acceleration effect but reflects a

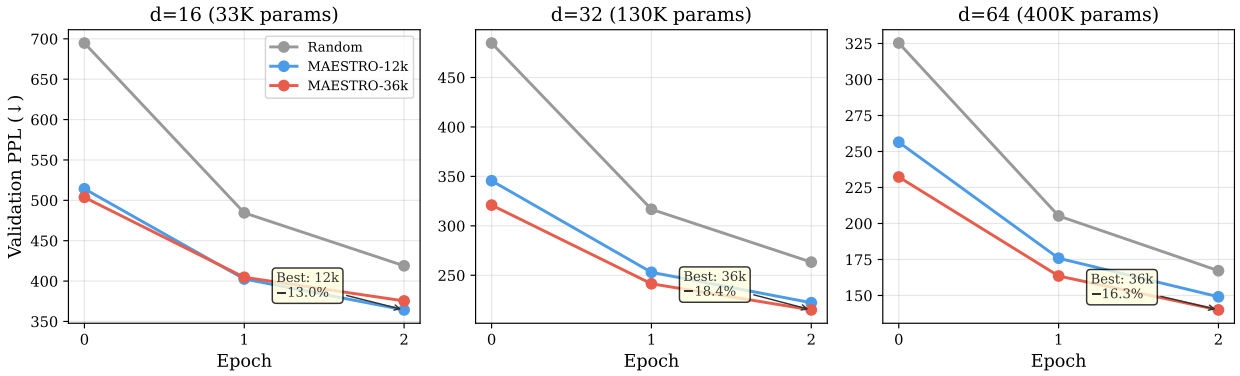

Figure 3: Learning curves across scales. At $d = 16$, MAESTRO-12k (blue) is the best condition; at $d = 32$ and $d = 64$, MAESTRO-36k (red) overtakes 12k, with the gap widening at larger scales. The annotation shows the best music condition and its improvement over the random baseline at epoch 2.

genuinely lower loss basin.

Second, **the gap varies across seeds but is consistently positive**. Individual gaps range from 1.5% to 8.9%, reflecting seed-dependent variation in how effectively the transferred structures are exploited during language learning. Despite this variation, the pipeline wins in every seed, and the mean effect is statistically significant.

Third, **the pipeline converges faster**. In every seed, the pipeline reaches plateau in fewer epochs than the random baseline (6–8 vs. 8–9). The developmental pipeline thus provides both a better destination and a faster path to reach it.

Note that the convergence gap (5.5% at $d=64$) is smaller than the three-epoch gap reported in Phase 2 (11.8–17.5%). This is expected: additional training epochs benefit both conditions, but the random baseline has more room for improvement from its higher starting point. The persistent gap at plateau represents the *irreducible* advantage of musical pre-training—structural knowledge that cannot be recovered by training longer on text alone.

5 Analysis

The preceding results establish *what* happens; here we interpret *why*, organized around six themes.

5.1 Why Does Music Transfer to Language?

The transferred attention weights carry computational structures—long-range dependency tracking, hierarchical pattern recognition—that directly benefit language processing. This is consistent with Lee et al. [2026]'s finding that attention weights are the most transferable component, and extends it by showing that music provides a particularly effective source domain. Table 10 summarizes the structural parallels that may explain this transfer.

5.2 Quality as Efficiency

MAESTRO-12k and Synth-36k reach statistically indistinguishable ceilings (Section 4.3), so the quality advantage of real music is best characterized as an *efficiency* advantage: denser structural information per

Table 10: Structural parallels between music and language.

Structure	Music	Language
Hierarchy	note → phrase → section	word → clause → paragraph
Long-range dep.	theme → development → recap.	subject–verb agreement
Expectation	dissonance → resolution	garden-path correction
Directionality	tension → resolution	given → new information

data point, enabling the model to saturate sooner. Synthetic data, being structurally simpler, requires more examples to extract the same amount of transferable structure.

5.3 Orthogonal Contributions Explain Additivity

Music and poetry improve *different model components*, and this separation is a direct consequence of the transfer design (Section 3.4). When transitioning from music to poetry, only vocabulary-independent layers—attention, feedforward, and layer normalization weights—are transferred; the token embedding and language model head are reinitialized because music tokens and language tokens have no correspondence. Consequently, any benefit of music pre-training is confined *by construction* to internal computation (attention + FFN), and cannot reside in embeddings.

The epoch-0 perplexities confirm this decomposition. After music-only transfer (MAESTRO-12k → Prose), the initial perplexity is ∼500—still high because the reinitialized embeddings carry no language information—yet subsequent learning proceeds significantly faster than the random baseline (−11.8% at epoch 2). This pattern is the signature of improved internal computation: the model does not *start* better, but *learns* faster. After the poetry phase (MAESTRO → Poetry → Prose), the initial perplexity drops to 416 ± 4, already below the random baseline's *converged* value (423.0). Since the internal weights are unchanged between the poetry and prose phases (same vocabulary, full model continues), this additional drop is attributable to embedding calibration.

The near-additivity of these effects further supports orthogonality: music alone improves perplexity by 11.8%, the poetry phase adds a further 5.7 percentage points (17.5% total), and this increment is consistent across all five seeds. If music and poetry competed for the same model capacity, we would expect sub-additive gains; instead, the effects compose because they target non-overlapping parameters.

5.4 A Developmental Ordering

The optimal training order—music → poetry → prose—mirrors a well-documented pattern in human infant development: sensitivity to pre-linguistic regularities (rhythm, prosody; 0–6 months) → phonemic discrimination (6–12 months) → vocabulary and grammar (12+ months) [Kuhl, 2004]. We suggest this is not coincidence: acquiring a primitive sense of regularity before linguistic content is a fundamentally efficient strategy, and biological development has converged on this solution through evolutionary pressure. Both systems face the same computational problem—bootstrapping language comprehension from sub-linguistic pattern recognition—and both benefit from the same curriculum: regularity first, language second. This principle generalizes curriculum learning [Bengio et al., 2009] by grounding the ordering not in task difficulty but in the progression from non-linguistic to linguistic structure.

5.5 Capacity-Dependent Data Curation

The monotonic shift of optimal data volume with model capacity (Section 4.4) has a practical implication: **pre-training data volume should be calibrated to model size**. Unlike language pre-training, where more data is almost always beneficial [Hoffmann et al., 2022], music pre-pre-training has a *capacity-dependent* optimum—the purpose is not to acquire domain knowledge (which scales with data) but to establish computational structures (which saturate once the model has learned the relevant patterns).

This extends Lee et al.'s observation that optimal data *complexity* varies by target domain. Our result adds a second axis: optimal *volume* varies by model capacity, defining a two-dimensional design space that should be navigated as a function of both the target domain and the model's absorptive capacity.

5.6 Irreducible Transfer

The convergence tests (Section 4.5) show that the pipeline's advantage is not a transient head start but an *irreducible* benefit. At $d = 64$, multi-seed validation confirms a mean 5.5% gap at plateau ($p = 0.017$), with the pipeline winning in all five seeds. This transforms the interpretation of the three-epoch results. The 11.8–17.5% improvements are the early manifestation of a persistent advantage that narrows as both conditions approach their respective ceilings, but never vanishes. Music pre-training provides structural knowledge that *cannot be recovered* by training longer on text alone.

6 Limitations

Our study has several limitations. First, while Phase 3 extends our findings to $d=64$ ($\sim$400K parameters), validation at substantially larger scales (millions of parameters) remains needed. Second, we evaluate only on English with classical Western music; generalization to other languages and musical traditions is unknown. Third, we report only perplexity; downstream task evaluation would strengthen the findings. Fourth, the music-to-language transfer requires vocabulary reinitialization, which discards information; more sophisticated transfer methods (*e.g.*, shared subword-MIDI vocabularies) may improve results. Fifth, while Phase 2 validates the main findings across five random seeds, the music checkpoints themselves use a single seed (seed 42), and Phase 3 (the scale experiment) uses a single seed throughout; multi-seed replication of the scaling results would strengthen the conclusions. Finally, the $d=16$ convergence test uses a single seed; while the $d=64$ convergence test is validated across five seeds ($p = 0.017$), multi-seed replication at $d=16$ would further strengthen the cross-scale comparison.

7 Conclusion

We have demonstrated that music pre-pre-training accelerates language learning in Transformers, with the full developmental pipeline achieving a 17.5% perplexity improvement ($p < 0.001$). Six findings emerge from our experiments: (1) real music by master composers provides a highly effective pre-training substrate, achieving the same transfer as synthetic patterns with one-third the data; (2) a developmental pipeline—music $\rightarrow$ poetry $\rightarrow$ prose—yields additive improvements that no single phase can match; (3) music and poetry improve orthogonal model components—internal computation and embeddings, respectively; (4) the poetry phase is so effective that the model enters prose training with perplexity already below the untrained baseline's final converged value; (5) optimal pre-training data volume scales with model capacity,

with the advantage of larger datasets growing monotonically from $d = 16$ through $d = 64$; and (6) the pipeline's advantage persists at convergence—multi-seed validation at $d{=}64$ confirms a 5.5% gap at plateau ($p = 0.017$), with the pipeline winning in every seed, confirming that the transferred structures provide an irreducible benefit rather than a transient head start. Together, these results suggest a developmental pipeline for language models—*listen, chant, then read*—grounded in the same progression that characterizes human language acquisition, with data volume calibrated to the model's absorptive capacity.

More broadly, our findings suggest that the quality of pre-training data—specifically, whether it is a product of skilled human creative activity—matters for efficiency: real music encodes more transferable structure per data point than synthetic alternatives. The capacity-dependent scaling of optimal data volume further implies that pre-training data curation should be treated as a joint function of data quality, data quantity, and model size.

8 Future Work

Several directions merit investigation. First, validating the developmental pipeline at larger scales ($d{=}128$, $d{=}256$) with a *FLOPs-matched* comparison against Wikipedia-only training would establish whether the pipeline's efficiency advantage holds when computational budgets are equalized at practical model sizes. Second, the structural similarity between our capacity-constrained setting and low-rank adaptation (LoRA) suggests that *developmental initialization*—warming up LoRA's low-rank subspace with structured data before domain fine-tuning—may improve fine-tuning efficiency, particularly at low ranks ($r = 4$–8) where the subspace direction matters most. Third, extending the pipeline to non-Western music and non-English languages would test whether the transfer relies on universal structural properties or culture-specific patterns. Finally, mapping the full interaction surface—music data volume × poetry data volume × model capacity—would enable principled curriculum design for the developmental pipeline at any target scale.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009.

BIG LAM. Gutenberg poetry corpus. `https://huggingface.co/datasets/biglam/gutenberg-poetry-corpus`, 2022.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In *International Conference on Learning Representations*, 2019. URL `https://magenta.tensorflow.org/datasets/maestro`.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35:30016–30030, 2022.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2019.

Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. *Proceedings of ACM Multimedia*, 2020.

Stefan Koelsch. Toward a neural basis of music perception – a review and updated model. *Frontiers in Psychology*, 2:110, 2011.

Patricia K. Kuhl. Early language acquisition: Cracking the speech code. *Nature Reviews Neuroscience*, 5 (11):831–843, 2004.

Dan Lee, Seungwook Han, Akarsh Kumar, and Pulkit Agrawal. Training language models via neural cellular automata. *arXiv preprint arXiv:2603.10055*, 2026.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *International Conference on Learning Representations*, 2017.

Aniruddh D. Patel. Language, music, syntax and the brain. *Nature Neuroscience*, 6(7):674–681, 2003.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. MusicBERT: Symbolic music understanding with large-scale pre-training. In *Findings of ACL*, 2021.